

### **IN THE CLAIMS**

Please cancel claim 12, amend claim 32 and add new claim 34 as set forth below.

The text of all pending claims, along with their current status, is set forth below:

1. (Previously presented) A chip-multiprocessing system with scalable architecture, comprising on a single chip:
  - a plurality of processor cores;
  - a two-level cache hierarchy including a pair of instruction and data caches for, and private to, each processor core, the pair being first level caches;
  - a second level cache with a relaxed inclusion property, the second-level cache being logically shared by the plurality of processor cores, the second level cache being modular with a plurality of interleaved modules;
  - one or more memory controllers capable of operatively communicating with the two-level cache hierarchy and with an off-chip memory;
  - a cache coherence protocol;
  - one or more coherence protocol engines;
  - an intra-chip switch; and
  - an interconnect subsystem.
  
2. (Original) A chip-multiprocessing system as in claim 1, wherein the scalable architecture is targeted at parallel commercial workloads.

3. (Original) A chip-multiprocessing system as in claim 1, further comprising on a single I/O chip (input output chip):

a processor core similar in structure and function to the plurality of processor cores;

a single-module second-level cache with controller;

an I/O router; and

a memory that participates in the cache coherence protocol.

4. (Original) A chip-multiprocessing system as in claim 1, wherein the plurality of core processors are each a single-issue, in-order processor configured with a pipelined datapath and hardware support for floating-point operations.

5. (Original) A chip-multiprocessing system as in claim 1, wherein the plurality of processor cores are each capable of executing an instructions set of the ALPHA™ processing core.

6. (Original) A chip-multiprocessing system as in claim 1, wherein the plurality of processor cores are each configured with a branch target buffer, pre-compute logic for branch conditions, and a fully bypassed datapath.

7. (Original) A chip-multiprocessing system as in claim 1, wherein each of the plurality of processor cores is capable of separately interfacing with either of the instruction and data caches, and wherein each of the caches is configured for single-cycle latency.

8. (Original) A chip-multiprocessing system as in claim 1, wherein the interconnect subsystem includes a network router, a packet switch and input and output queues.

9. (Original) A chip-multiprocessing system as in claim 1, wherein the single chip creates a node, and wherein the coherence protocol engines include a home engine and a remote engine which support shared memory across multiple nodes.

10. (Original) A chip-multiprocessing system as in claim 1, further comprising:  
a system control module that takes care of system initialization and maintenance  
including configuration, interrupt handling, and performance monitoring.

11-12. (Canceled)

13. (Original) A chip-multiprocessing system as in claim 1, wherein the second level cache is interleaved into eight modules.

14. (Original) A chip-multiprocessing system as in claim 1, wherein each of the instruction and data caches is a two-way set-associative, blocking cache with virtual indices and physical tags.

15. (Original) A chip-multiprocessing system as in claim 1, wherein each instruction cache is kept coherent by hardware.

16. (Original) A chip-multiprocessing system as in claim 1, wherein each of the second level cache modules includes an N-way set associative cache and uses a round-robin or least-recently-loaded replacement policy if an invalid block is not available.

17. (Canceled)

18. (Original) A chip-multiprocessing system as in claim 1, wherein the pair of instruction and data caches includes a first state field per each cache line present therein the first state field having bits related to the MESI (modified, exclusive, shared, invalid) protocol.

19. (Original) A chip-multiprocessing system as in claim 18, wherein the second level cache maintains a duplicate of the first state fields from the first-level pairs of instruction and data caches, the duplicate being maintained in order to avoid the need for a first-level cache lookup for cache lines that map to given addresses of corresponding requested cache lines.

20. (Original) A chip-multiprocessing system as in claim 18, wherein the second level cache holds a second state field for each cache line present therein, the second state field having bits related to the MESI protocol, wherein the second level cache maintains a duplicate of the first state fields, and wherein on every second level cache access the duplicate first state fields and the second state fields are accessed in parallel.

21. (Original) A chip-multiprocessing system as in claim 1, wherein the single chip creates a node, and wherein information about sharing of data across nodes is kept in a directory in a memory accessed via the memory controllers.

22. (Original) A chip-multiprocessing system as in claim 21, wherein the second level cache includes a controller, and wherein manipulation and interpretation of the directory is done by the protocol engines, although the controller also interprets the directory, but merely for determining whether a cache line is cached remotely to the single chip.

23. (Original) A chip-multiprocessing system as in claim 1, wherein the interconnect subsystem includes at least one datapath, and wherein the interconnect subsystem is a crossbar configured with a uni-directional, push-only interface, and is capable of scheduling data transfers according to datapaths availability, pre-allocating datapaths, speculatively asserting a requester's grant signal, and supporting back-to-back transfers without dead-cycles between transfers.

24-25. (Canceled)

26. (Original) A chip-multiprocessing system as in claim 1, wherein the memory controller includes a memory access controller with high speed interface circuitry and a memory controller engine capable of scheduling second-level cache memory access.

27. (Original) A chip-multiprocessing system as in claim 1, wherein the coherence protocol engines are implemented as similarly structured microprogrammable controllers, although each of them has its respective microcode.

28. (Original) A chip-multiprocessing system as in claim 1, wherein each of the coherence protocol engines is configured with an input stage, a microcode-controlled execution stage and an output stage.

29. (Original) A chip-multiprocessing system as in claim 1, wherein at least one of the coherence protocol engines is configured to execute protocol code that includes instructions named Send, Receive, Lsend, Lreceive, Test, Set and Move.

30. (Previously presented) A method for scalable chip-multiprocessing, comprising:

- (a) providing on a single chip
  - (i) a plurality of processor cores,
  - (ii) a two-level cache hierarchy including
    - (A) a pair of instruction and data caches for, and private to, each processor core, the pair being first level caches, and
    - (B) a second level cache with a relaxed inclusion property, the second-level cache being logically shared by the plurality of processor cores, the second level cache being modular with a plurality of interleaved modules,

- (iii) one or more memory controllers capable of operatively communicating with the two-level cache hierarchy and with an off-chip memory,
  - (iv) a cache coherence protocol,
  - (v) one or more coherence protocol engines,
  - (vi) an intra-chip switch, and
  - (vii) an interconnect subsystem,
- (b) wherein the single chip creates a node; and
- (c) providing one or more than one of the nodes to create, in a modular scalable fashion, a glueless multiprocessor.

31. (Original) A method for scalable chip-multiprocessing as in claim 30, further comprising:

providing on a single I/O chip (input output chip)

a processor core similar in structure and function to the plurality of processor cores,

a single-module second-level cache with controller,

an I/O router, and

a memory that participates in the cache coherence protocol.

32. (Currently amended) A single-chip multiprocessing system, comprising:

a plurality of processor cores;

a two-level cache hierarchy including a pair of instruction and data caches for, and private to, each processor core, the pair being first level caches;

a second level cache with a relaxed inclusion property that is logically shared by the plurality of processor cores, the second level cache being modular with a plurality of interleaved modules; and

a plurality of memory controllers, each of the plurality of memory controllers being associated with one of the plurality of interleaved modules, each of the plurality of the memory controllers being adapted to communicate with the two-level cache hierarchy and with an off-chip memory.

33. (Previously presented) The single-chip multiprocessing system set forth in claim 32, wherein each of the plurality of interleaved modules of the second level cache comprises dedicated tag and data storage.

34. (New) A chip-multiprocessing system, comprising:

a plurality of processor cores;

a two-level cache hierarchy including a pair of instruction and data caches for, and private to, each processor core, the pair being first level caches;

a second level cache with a relaxed inclusion property, the second-level cache being logically shared by the plurality of processor cores, the second level cache being modular with a plurality of interleaved modules, wherein each of the plurality of interleaved modules of the second level cache has its own controller, on chip tag and data storage, and wherein each module being attached to one of the memory controllers which interfaces to a bank of memory chips, and each bank of memory chips includes DRAM (dynamic random access memory) chips;



one or more memory controllers capable of operatively communicating with the two-level  
cache hierarchy and with an off-chip memory;  
a cache coherence protocol;  
one or more coherence protocol engines;  
an intra-chip switch; and  
an interconnect subsystem.